SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 3706	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A USER-ORIENTED MANAGEMENT PROGRAM FOR PROJECT MONITORING ON A TIME-SHARE COMPUTER TERMINAL		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Clifford L. Temes		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D. C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem R02-92 Project ZF-12-151-001
11. CONTROLLING OFFICE NAME AND ADDRESS CND Direct Funded Programs Naval Research Laboratory Washington, D.C. 20375		12. REPORT DATE January 1978 13. NUMBER OF PAGES 21
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT *(of this Report)* Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)* Project management		
20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)* This report describes a (Fortran) management computer program for monitoring technical progress and expenditures on a time-share terminal. The program is written for use on the DEC-10 computer at NRL. A basic feature of the output is a plot of actual past expenditures combined with estimated future expenditures, along with other project information such as technical milestones, major procurements and staffing.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

NRL Memorandum Report 3706

A User-Oriented Management Program for Project Monitoring on a Time-Share Computer Terminal

CLIFFORD L. TEMES

Search Radar Branch
Radar Division

January 1978

NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

A USER-ORIENTED MANAGEMENT PROGRAM FOR PROJECT MONITORING ON A TIME-SHARE COMPUTER TERMINAL

INTRODUCTION

This report describes a Fortran time-share computer program developed and used by the author to monitor technical progress and expenditures on a number of simultaneous projects in the Search Radar Branch, Radar Division. The program design, and its description here, are user-oriented for ready adaptation by the reader. The program is written for the DEC-10 computer at NRL. A basic feature of the output is a plot of actual past expenditures combined with estimated future expenditures, along with other project information such as technical milestones, major procurements and staffing. The program covers a single fiscal year from 1 October through 30 September.

Another NRL management program* known to the author is also available. That program differs from the one described here in that it is executed by batch processing on the ASC, uses a CALCOMP plotter and does not include milestones (although the latter would be a minor addition). NSWC/WO is also understood to have an interactive management program, but information on that program has not yet been received by the author.

SAMPLE OUTPUT

Figure 1 shows a sample of a complete program output, although much of the sample printout may be omitted at the option of the user by appropriately answering the terminal's interrogations in the negative when it offers particular output segments.

As shown on the printout, the program is executed by using a compound command such as:

```
EX MANAGE.FOR,R0224.FOR
```

where

MANAGE.FOR is the name of a general main-body program

R0224.FOR is the name of a specific subprogram** applying only to the project R02-24.

---

*T.H. Gauss, "A Fortran Routine to Plot Data from the NRL Job Order Status Reports," NRL Document 527856, 22 April 1977.

**Separately storing the major block of code (MANAGE.FOR) common to all runs saves substantial disk storage.

Note: Manuscript submitted January 16, 1978.

```
.EX MANAGE.FOR;R0224.FOR
LINK:   LOADING
[LNKXCT MANAGE EXECUTION]
LIST OF MILESTONES? YES=1;NO=0
1

1. COMPLETION & DELIV. OF TRAKX TO PAX R.  (2;28)
2. PRELIM. EVAL. OF TRAKX (10;31;78)
3. INSTALLATION OF PULSE COMPRESS. DEVICE COMPLETED (2;28;79)
4. EVAL. OF PULSE COMPRESS. DEVICE COMPLETED (6;30;79)
5. REPT. ON INITIAL PERFORM. COMPLETED (9;30;79)
6. MEASUREMENTS ESTABLISHING VECTOR MISS DISTANCE (6;30;80)
7. COMPLETE REPT. ON PERFORMANCE (9;30;80)

INPUT:PRESENT DATE--M;D
10;30

INPUT:EXPEND;OBLIG.;ENCUMB.;EXTRA
15431;0;0;400


*=BUDGET FUNDING;  .=PLANNED SPENDING
+=AVAILABLE FUNDS;  0=EXPEND. & COMMIT.
 125000.+    *   *   *   *   *   *   +   +   +   +   +   +   +
         I                                                   0
         I                                                   .
         I                                                   0
 104167.+                                                   .
         I                                                   0
         I                                               .
         I                                               0
         I
  83333.+                                           0
         I
         I   +   +   +   +   +   +       0
         I
         I
  62500.+                           0
         I
         I                       0
         I
         I                   0
  41667.+                   .
         I           0
         I           .
         I
         I       0
  20833.+       .
         I   0
         I   .
         I
         I
      0.+  0
           _   _   _   _   _   _   _   _   _   _   _   _   _
           9  10  11  12   1   2   3   4   5   6   7   8   9
```

Fig. 1

```
LIST OF MAJOR PROCUREMENTS? YES=1,NO=0
1
~
NO MAJOR PROCUREMENTS

PROJECT STAFF? YES=1,NO=0
1

HOWARD(.3),ORNSTEIN(.3),CROSS(.3),LIPKA(.2),MAYS(.3),J.WARD(.4)


LIST OF NUMERICAL VALUES ON GRAPH? YES=1,NO=0
1
```

NO.	*	.	+	0	BAL.(+-0)
9	125000.	0.	75000.	0.	75000.
10	125000.	10417.	75000.	15431.	59569.
11	125000.	20833.	75000.	24675.	50325.
12	125000.	31250.	75000.	33919.	41081.
1	125000.	41667.	75000.	43163.	31837.
2	125000.	52083.	75000.	52407.	22593.
3	125000.	62500.	125000.	61651.	63349.
4	125000.	72917.	125000.	70895.	54105.
5	125000.	83333.	125000.	80139.	44861.
6	125000.	93750.	125000.	89383.	35617.
7	125000.	104167.	125000.	98627.	26373.
8	125000.	114583.	125000.	107871.	17129.
9	125000.	125000.	125000.	117115.	7885.

```
END OF EXECUTION
CPU TIME: 4.56  ELAPSED TIME: 3:27.62
EXIT

.K/F
JOB 2, USER [536,1333]  LOGGED OFF TTY14     1218  22-NOV-77
SAVED ALL FILES (240 BLOCKS) RUNTIME 7 SEC    COST (APPROX) $0.18
CONNECT TIME 4 MIN 10 SEC    COST (APPROX) $0.38
KCS: 98  COST (APPROX) $0.04
TOTAL COST FOR JOB (APPROX.) $0.60
```

Fig. 1 — Sample of complete program output

At the beginning of the printout, the terminal asks the user if a list of milestones is wanted. In the sample run, the user replied with a 1 (for yes) and the list of milestones was printed, with each milestone being followed, in parenthesis, by a target completion date, (month, day) for the current fiscal year and (month, day, year) for dates beyond the current fiscal year.

The terminal then asks for the current date of the run (month, day). The program was designed to be run using the last day of each nominal thirty-day month as an input here (nominally in a form such as "2,30" for February, etc.), although the actual run would be made at a later date when the expenditure information printout (Job Order Status Report) becomes available from the NRL Budget Office. The information from the Job Order Status Report is entered at the next terminal interrogation in the form of four input numbers: Expenditures, Obligations, Encumbrances, and Extra. The latter, which is not self-explanatory, is an expenditure amount to be added to each month of the future extrapolation to account for average travel, ESD support, small procurements, etc., which are not the two major components of the extrapolation, namely (1) labor and (2) major procurements.

After receiving the four inputs (Expenditures, Obligations, Encumbrances and Extra), the terminal plots: (1) a curve of *'s along the top border, representing the Budget Funding; (2) a curve of +'s, representing the buildup of funds received from the sponsor up to the current date or expected in installments from the sponsor at specified future dates;* (3) a curve of .'s, representing the planned spending that includes the major procurements at specified dates, with the remainder of the spending being linearly interpolated to exactly use up the Budget Funding amount and (4) a curve of 0's, representing actual spending (Expenditures + Obligations + Encumbrances) up to the current date and extrapolated spending estimates for the future, taking into account the scheduled major procurements, the labor assigned to the job and the EXTRA amount added each month.

Following the graph, the user is interrogated with regard to whether a list of major procurements is desired. If answered affirmatively, the list is printed out with the name, date and cost of each procurement.

The next interrogation of the user deals with the identification of the project staff scheduled for the job. If answered affirmatively, the list of staff members is printed along with the fraction of time to be spent on the project.

Finally, the terminal queries the user on whether the numerical values plotted on the graph should be printed out. An affirmative answer results in the printed values along with a running balance showing the difference between the available funds and the expenditures (actual to current date and extrapolated thereafter). The lower-most right-hand number gives the estimated surplus at the end of the fiscal year.

---

*In Fig. 1, the + curve has replaced the * curve along some of the top border.

As shown on the sample printout, the cost of a run can be less than $1.00 if no compilation is required. With compilation (and no disk storage charge for a compiled version of the program), the run-cost of the job shown was $1.05 instead of the $.60 indicated on the printout.

SETTING UP A PROJECT ON THE COMPUTER PROGRAM

The computer program consists of four parts:

(1) A general Fortran file, called MANAGE.FOR, which is used for all projects.

(2) A specific Fortran file for each project, designated by a name such as R0224.FOR, constructed from a project number such as R02-24; this contains project data.

(3) A specific short file for each project (consisting of 13 numbers), designated by a name such as SR0224.DAT, again constructed from the project number, but preceded by an "S"; this file consists of the (actual and extrapolated) monthly expenditures that are plotted by the symbol "0" on the graph and is updated each month during a regular program run.

(4) A general file called PSAL.DAT which contains personnel weekly salaries referenced by personnel ID numbers.

A listing of the general Fortran file (Item (1)) is given in Appendix A and need not be altered for running with different projects.

A listing of a sample specific project Fortran file is given in Appendix B. One of these is needed for each project. The project data information in the listing is as follows:

Subroutine Data

NAME = storage file associated with a specific project, e.g. SR0292.DAT

BF = total budget funding for project

AF = available funds received from sponsor at start of fiscal year, 1 October

IAFE(I,1), IAFE(I,2)* = month, day on which the Ith installment is to be received for available funds expected

AFE(I) = amount of the Ith installment for available funds expected

*For all matrix inputs, note dimensions on listing for maximum number of items allowed in present code. Dimensions may be modified as desired by the user.

IP(I,1), IP(I,2) = month, day on which the Ith major procurement is to be charged

P(I) = amount of Ith major procurement

IPID (I) = ID number of Ith man on job

PID (I) = fraction of time on job for Ith man

Subroutine Mile = a list of milestone names and dates

Subroutine Pro = a list of major-procurement names, dates and amounts

Subroutine Staff = a list of staff names and fraction of time on job.

To set up the first job, the file PSAL.DAT must be created by running a program such as INF1.FOR, shown in Appendix C. This loads the file PSAL.DAT with salaries (per manweek) in order of personnel ID numbers*. The salaries are taken from a table of the type shown in Appendix D, supplied by Division Offices, which includes nominal average salary, G&A and Division Indirect costs as a function of GS grade, as charged to the contract by the central Budget Office.

Then a project data file (e.g., SR0224.DAT) must be created by running the program INF.FOR, shown in Appendix E.

With all four of the above-described programs available in the computer, a run can be executed using the compound command

EX MANAGE.FOR,R0224.FOR

MONTHLY USE OF THE PROGRAM

The program assumes that each month consists of thirty days. After expenditure information for October is received from the NRL Budget Office, the first run of the program is made using the input date 10,30. The expenditure information is permanently stored in the second element of the data file (SR02 24.DAT), with the first element being a zero for the date 9,30 of the previous fiscal year.

The second run is made using the input date 11,30, and so forth for successive months. On each run the actual expenditures are permanently stored for the current month, as they were for the previous months, thus allowing the data file (SR02 24.DAT) to build up real expenditures for the past and present months and calculated expenditures for the future months. (This file is plotted by 0's on the output graph.)

*The Branch personnel are simply indexed in order, to provide these Branch ID numbers.

APPENDIX A

LISTING OF GENERAL FORTRAN FILE, MANAGE.FOR

```
TYPE MANAGE.FOR
00200           DIMENSION P(15),IP(15,2)
00210           DIMENSION AFE(6),IAFE(6,2),IPID(30),PID(30)
00300           DIMENSION IDATE(2)
00350           DOUBLE PRECISION NAME
00400           DIMENSION ARRAY(13,4),PS(35)
00410           CALL DATA(BF,AF,P,IP,AFE,IAFE,IPID,PID,NAME)
05700           OPEN(UNIT=21,DEVICE='DSK',FILE='PSAL.DAT',ACCESS='SEQIN')
05800           READ(21,14)  (PS(L),L=1,35)
05900        14 FORMAT(35F5.0)
06000           CLOSE(UNIT=21,DEVICE='DSK',FILE='PSAL.DAT')
06200           OPEN(UNIT=20,DEVICE='DSK',FILE=NAME,ACCESS='SEQIN')
06300           READ(20,6)  (ARRAY(I,4),I=1,13)
06400         6 FORMAT(13F10.0)
06500           CLOSE(UNIT=20,DEVICE='DSK',FILE=NAME)
06600           WRITE(5,16)
06700        16 FORMAT(1X,'LIST OF MILESTONES? YES=1,NO=0')
06800           ACCEPT*,IANS
06900           IF(IANS.EQ.1) CALL MILE
07000           WRITE(5,4)
07100        4  FORMAT(/1X,'INPUT:PRESENT DATE--M,D')
07200           ACCEPT*,IDATE(1),IDATE(2)
07300           WRITE(5,1)
07400         1 FORMAT(/1X,'INPUT:EXPEND,OBLIG.,ENCUMB.,EXTRA')
07500           ACCEPT*,EX,OB,EN,EXTRA
07600           CE=EX+OB+EN
07700           SUM=0.
07800           DO 100 I=1,15
07900        100 SUM=SUM+P(I)
08000           SLOPE=(BF-SUM)/12.
08100           IBB=IDAYS(IDATE(1),IDATE(2))
08200           VAR=IBB/30
08300           MO=IFIX(VAR)+1
08400           ARRAY(1,1)=BF
08500           ARRAY(1,2)=0.
08600           ARRAY(1,3)=AF
08700           ARRAY(1,4)=0.
08800           CALL SCANP(IPID,PID,PS,COMSAL)
08900           DO 110 I=2,13
09000           IX=(I-1)*30
09100           ARRAY(I,1)=BF
09200           CALL SCAN1(IX,P,IP,DELP)
09300           ARRAY(I,2)=ARRAY(I-1,2)+SLOPE+DELP
09400           CALL SCAN2(IX,AFE,IAFE,DELE)
09500           ARRAY(I,3)=ARRAY(I-1,3)+DELE
09600           IF(I.EQ.MO)  ARRAY(I,4)=CE
09700           IF(I.GT.MO)  ARRAY(I,4)=ARRAY(I-1,4)+COMSAL+DELP+EXTRA
09800           OPEN(UNIT=20,DEVICE='DSK',FILE=NAME,ACCESS='SEQOUT')
09900           WRITE(20,6)  (ARRAY(II,4),II=1,13)
10000           CLOSE(UNIT=20,DEVICE='DSK',FILE=NAME)
10100        110 CONTINUE
10200           CALL GRAPH(BF,ARRAY)
10300           WRITE(5,18)
10400        18 FORMAT(//1X,'LIST OF MAJOR PROCUREMENTS? YES=1,NO=0')
10500           ACCEPT*,IANS
10600           IF(IANS.EQ.1) CALL PRO
10700           WRITE(5,21)
10800        21 FORMAT(/1X,'PROJECT STAFF? YES=1,NO=0')
```

```
10900          ACCEPT*,IANS
11000          IF(IANS.EQ.1) CALL STAFF
11100          WRITE(5,19)
11200       19 FORMAT(//1X,'LIST OF NUMERICAL VALUES ON GRAPH? YES=1,NO=0')
11300          ACCEPT*,IANS
11400          IF(IANS.EQ.1) WRITE(5,9)
11500       9  FORMAT(//3X,'MO.',10X,'+',14X,'.',14X,'+',14X,
11510         1 '0',10X,'BAL(+-0)'//)
11600          IF(IANS.EQ.0) GO TO 130
11700          DO 140 I=1,13
11710          BAL=ARRAY(I,3)-ARRAY(I,4)
11800          IND=I-4
11900          IF(I.LE.4) IND=I+8
12000          WRITE(5,8) IND,(ARRAY(I,J),J=1,4),BAL
12100       8  FORMAT(3X,I2,4(5X,F10.0),5X,F10.0)
12200       140 CONTINUE
12300       130 CONTINUE
12400          END
12500          SUBROUTINE SCAN1(IX,Q,IQ,DEL)
12600          DIMENSION Q(15),IQ(15,2)
12700          DEL=0.
12800          DO 100 I=1,15
12900          IZ=IDAYS(IQ(I,1),IQ(I,2),IQ(I,3))
13000       100 IF(IZ.LE.IX.AND.IZ.GT.IX-30) DEL=DEL+Q(I)
13100          RETURN
13200          END
13300          SUBROUTINE SCAN2(IX,Q,IQ,DEL)
13400          DIMENSION Q(6),IQ(6,2)
13500          DEL=0.
13600          DO 100 I=1,6
13700          IZ=IDAYS(IQ(I,1),IQ(I,2),IQ(I,3))
13800       100 IF(IZ.LE.IX.AND.IZ.GT.IX-30) DEL=DEL+Q(I)
13900          RETURN
14000          END
14100          SUBROUTINE SCANP(IPID,PID,PS,COMSAL)
14200          DIMENSION IPID(30),PID(30)
14300          DIMENSION PS(35)
14400          W=0.
14500          DO 100 I=1,30
14600          M=IPID(I)
14700          SAL=PS(M)
14800          W=W+SAL*PID(I)
14900       100 CONTINUE
15000          COMSAL=W*45./12.
15100          RETURN
15200          END
15300          INTEGER FUNCTION IDAYS(M,ID)
15400          IF(M.LE.9) MP=M+2
15500          IF(M.GT.9) MP=MOD(M+2,12)
15600          IDAYS=MP*30+ID
15700          RETURN
15800          END
15900          SUBROUTINE GRAPH(BF,ARRAY)
16000          DIMENSION ARRAY(13,4)
16100          DIMENSION PLOT(32,15)
16200          DIMENSION KPLOT(15)
16300          IFLAG=0
16400          ISCAL=30
16500          DO 100 I=1,ISCAL
16600          DO 100 J=1,15
16700          PLOT(I,J)=1H
```

```
16800      100 CONTINUE
16900          DO 110 K=1,ISCAL+1
17000          PLOT(K,2)=1HI
17100          IF(MOD(K-1,5).EQ.0) PLOT(K,2)=1H+
17200          IF(MOD(K,5).EQ.1) IFLAG=IFLAG+1
17300          IF(MOD(K,5).EQ.1) PLOT(K,1)=(BF/6)*(7-IFLAG)
17400      110 CONTINUE
17500          DO 130 J=3,15
17600          I1=(ISCAL+1)-ARRAY(J-2,1)*ISCAL/BF
17700          I2=(ISCAL+1)-ARRAY(J-2,2)*ISCAL/BF
17800          I3=(ISCAL+1)-ARRAY(J-2,3)*ISCAL/BF
17900          I4=(ISCAL+1)-ARRAY(J-2,4)*ISCAL/BF
18000          PLOT(I1,J)=1H*
18100          PLOT(I2,J)=1H.
18200          PLOT(I3,J)=1H+
18300          IF(I4.LT.1) GO TO 130
18400          PLOT(I4,J)=1H0
18500      130 CONTINUE
18600          DO 120 J=3,15
18700      120 PLOT(ISCAL+2,J)=1H-
18800          WRITE(5,14)
18900      14  FORMAT(//1X,'*=BUDGET FUNDING; .=PLANNED SPENDING')
19000          WRITE(5,15)
19100      15  FORMAT(1X,'+=AVAILABLE FUNDS; 0=EXPEND. & COMMIT.')
19200          DO 140 I=1,ISCAL+2
19300          IF(MOD(I-1,5).EQ.0) WRITE(5,10) (PLOT(I,L),L=1,15)
19400      10  FORMAT(1X,F8.0,14A4)
19500          IF(MOD(I-1,5).EQ.0) GO TO 140
19600          WRITE(5,11) (PLOT(I,L),L=1,15)
19700      11  FORMAT(1X,A8,14A4)
19800      140 CONTINUE
19900          DO 160 I=1,2
20000      160 KPLOT(I)=1H
20100          DO 170 I=3,15
20200          IF(I.LE.6) KPLOT(I)=I+6
20300          IF(I.GT.6) KPLOT(I)=I-6
20400      170 CONTINUE
20500          WRITE(5,12) (KPLOT(J),J=1,15)
20600      12  FORMAT(1X,A8,A2,13(1X,I2,1X))
20700          RETURN
20800          END
```

APPENDIX B

LISTING OF SPECIFIC PROJECT FORTRAN FILE, R0224.FOR

```
TYPE R0224.FOR
00100           SUBROUTINE DATA(BF,AF,P,IP,AFE,IAFE,IPID,PID,NAME)
00200           DIMENSION AFE(6),IAFE(6,2),IPID(30),PID(30)
00300           DIMENSION IP(15,2),P(15)
00400           DOUBLE PRECISION NAME
00500           NAME='SR0224.DAT'
00600           BF=125000.
00700           AF=75000.
00710           IAFE(1,1)=3; IAFE(1,2)=30
00720           AFE(1)=50000.
00800           IPID(1)=13;PID(1)=.3
00900           IPID(2)=22; PID(2)=.3
00910           IPID(3)=8; PID(3)=.3
00920           IPID(4)=17; PID(4)=.2
00930           IPID(5)=19; PID(5)=.3
00940           IPID(6)=33; PID(6)=.4
01000           RETURN
01100           END
01200           SUBROUTINE MILE
01300           WRITE(5,1)
01400      1    FORMAT(/1X,'1. COMPLETION & DELIV. OF TRAKX TO PAX R.
01500          1 (2,28)'/1X,'2. PRELIM. EVAL. OF TRAKX (10,31,78)'/1X,'3.
01510          2 INSTALLATION OF PULSE COMPRESS. DEVICE COMPLETED (2,28,79)'
01520          3/1X,'4. EVAL. OF PULSE COMPRESS. DEVICE COMPLETED (6,30,79)'
01530          4/1X,'5. REPT. ON INITIAL PERFORM. COMPLETED (9,30,79)'
01540          5/1X,'6. MEASUREMENTS ESTABLISHING VECTOR MISS DISTANCE
01550          6 (6,30,80)'/1X,'7. COMPLETE REPT. ON PERFORMANCE (9,30,80)')
02000           RETURN
02100           END
02200           SUBROUTINE PRO
02300           WRITE(5,1)
02400      1    FORMAT(/1X,'NO MAJOR PROCUREMENTS')
02500           RETURN
02600           END
02700           SUBROUTINE STAFF
02800           WRITE(5,1)
02900      1    FORMAT(/1X,'HOWARD(.3),ORNSTEIN(.3),CROSS(.3),LIPKA(.2),
02910          1MAYS(.3),J.WARD(.4)')
03000           RETURN
03100           END
```

APPENDIX C

LISTING OF PROGRAM INF1.FOR WHICH CREATES THE DATA FILE OF PERSONNEL SALARIES, PSAL.DAT

```
.TYPE INF1.FOR
00100           DIMENSION PS(35)
00200           OPEN(UNIT=21,DEVICE='DSK',FILE='PSAL.DAT',ACCESS='SEQOUT')
00300           PS(1)=1652.;PS(2)=1127.;PS(3)=1504.;PS(4)=1236.
00400           PS(5)=867.;PS(6)=1236.;PS(7)=1364.;PS(8)=1364.
00500           PS(9)=1364.;PS(10)=1236.;PS(11)=1031.;PS(12)=1236.
00600           PS(13)=1652.;PS(14)=1504.;PS(15)=1364.;PS(16)=1364.
00700           PS(17)=1364.;PS(18)=1364.;PS(19)=1236.;PS(20)=1504.
00800           PS(21)=1364.;PS(22)=1504.;PS(23)=1236.;PS(24)=1364.
00900           PS(25)=831.;PS(26)=932.;PS(27)=1364.;PS(28)=1364.
01000           PS(29)=1236.;PS(30)=1652.;PS(31)=1504.;PS(32)=1236.
01100           PS(33)=997.;PS(34)=1504.;PS(35)=1652.
01200           WRITE(21,1)  (PS(I),I=1,35)
01300         1 FORMAT(35F5.0)
01400           CLOSE(UNIT=21,DEVICE='DSK',FILE='PSAL.DAT')
01500           END
```

APPENDIX D

SAMPLE TABLE SHOWING CHARGED COSTS PER MANWEEK AS A FUNCTION OF GS GRADE LEVEL

August 1977

FY-1978

Man Week Rates

GS-1	745
GS-2	768
GS-3	797
GS-4	831
GS-5	867
GS-6	932
GS-7	966
GS-8	997
GS-9	1031
GS-10	1078
GS-11	1127
GS-12	1236
GS-13	1364
GS-14	1504
GS-15	1652

The above rates include pay, fringe benefits accumulation and applied G&A and Indirect overhead per man week.

G&A	436/week
Indirect	145/week
	581/week

APPENDIX E

LISTING OF PROGRAM INF.FOR WHICH CREATES THE PROJECT DATA FILE SR0224.DAT CONTAINING REAL EXPENDITURES FOR THE PAST AND PRESENT MONTHS AND CALCULATED EXPENDITURES FOR THE FUTURE MONTHS

```
TYPE INF.FOR
00100           DIMENSION X(13)
00200           DO 100 I=1,13
00300       100 X(I)=0.
00400           OPEN(UNIT=20,DEVICE='DSK',FILE='SR0224.DAT',ACCESS=SEQOUT')
00500          . WRITE(20,1)  (X(I),I=1,13)
00600       1   FORMAT(13F10.0)
00700           CLOSE(UNIT=20,DEVICE='DSK',FILE='SR0224.DAT')
00800           END
```

DEPARTMENT OF THE NAVY

NAVAL RESEARCH LABORATORY
Washington, D.C. 20375

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE NAVY
DoD-316